Classification in SQuAD 2.0 using BERT

Brian Lim

CS 291A - William Wang

Abstract

SQuAD 2.0 added the additional challenge to their Question Answering benchmark of including questions that are unable to be answered with the knowledge within the given context. This can be formulated as a classification problem. Google released a new model involving bidirectional transformers that performed extremely well on all benchmarks in all aspects of NLP. The main problem with BERT is it can only transform 512 tokens at a time. We propose a sliding window approach for long sequences with an optional added secondary sequence to provide additional information to combat BERT's restriction for long sequences, and specifically for SQuAD 2.0.

1. Introduction and Background

Question Answering (QA) is a specific MRC task that challenges computers to learn from textual evidence such as Wikipedia in order to answer reading comprehension questions derived directly from an information corpus. In contemporary media, QA has been popularized by IBM Watson, which is a machine that could play Jepoardy, a popular trivia game shown on television. Watson was able to comb through thousands of wikipedia articles to gather the answers to the Jepoardy question asked. In research, the current primary benchmark of QA is known as the Stanford Question Answer Dataset (SQuAD 1.1), which comprises of short excerpts from Wikipedia and questions where the answers are a short consecutive subsequence of the excerpts. The top scoring model for this dataset was created recently by Google using their new model consisting of Bidirectional Encoder Representations from Transformers (BERT). This new method has shown to be better than humans reading comprehension capability by a significant margin.

However, there is a concern that models that learn how to find answers to questions from textual evidence are overfitting to related keywords found in the question, thus focusing on words preceding or following those keywords, so Stanford has released a new challenge designed to be a benchmark for testing how accurate the information extracted from an information corpus is. This challenge is known as the Stanford Question Answer Dataset 2.0 (SQuAD 2.0). This challenge extends from the previous benchmark by introducing questions that are impossible to answer given the Wikipedia excerpt. In essence, we want to know whether computers can understand what they can and cannot find the answer to.

1.1. SQuAD 2.0

SQuAD 1.0 seemed to create models that learn context and type-matching heuristics, and fall victim to distracting sentences that do not provide answers to questions, but seem vaguely related to the question asked. They got the idea to create a new dataset, SQuAD 2.0, that would deal with this prevalent problem with current MRC models. In order to build a new dataset to find models that mitigate this tendency to overfit, they decided to create questions that are unanswerable. These questions follow one or two heuristics: they are relevant to the topic in the information except provided, and/or there are plausible answers to the question that follow a type-matching schema. Questions that follow these heuristics typically end up in one of several categories. Negation and Antonym questions ask the opposite of what the information excerpt states. Entity Swap questions are when an entity is replaced with one that sounds similar, but is not mentioned anywhere in the excerpt. Mutual Exclusion and Impossible Condition questions condition the question with an extra qualification that cant be answered based on the excerpt. Other unanswerable questions just simply do not have answers within the excerpt. They released this dataset as a challenge to see if models can adapt to more complex MRC tasks. Baseline models on this challenge score at around 65/67 EM/F1 score, which is 23 points below human performance. The hope of this challenge is for new models to be created that can understand human language at a deeper level by understanding what information is unknown and unable to be discerned from textual evidence. [1]

1.2. BERT

(Figure 1) Typically, languages are read left to right, so models tend to also follow the same direction. This method of left-to-right inputs sometimes doesnt capture the full meaning of a sentence since the meaning of earlier words in a sentence can depend on the meanings of future words. Humans can deal with that quirk of language quite efficiently, so reading left to right isnt a problem. Google created a new word embedding/model architecture, BERT. BERT is a dense neural network that uses dense bi-directional transformers to transform input word token representations into complex deep embeddings that extract features from the whole text and outputs those features as new word embeddings. This model can replace previous embedding models to improve their performance even more than ELMo has. However, Google did not try and integrate BERT with more complex models, and instead used it almost vanilla for many different benchmarks, scoring significantly higher than the previous best models. On SQuAD 1.0, they scored much higher than human performance, where previous state-of-the-art models could only score on par with humans. [2]

The largest drawback of BERT is that it has a limited input token sequence length, capped at 512. This means BERT by itself cannot encode large texts such as the contexts found in SQuAD. Methods will have to be created to deal with long sequences.

2. Sliding Window Approach

We introduce the Sliding Window Approach to BERT to deal with long sequences. This approach can also be used to use BERT on smaller GPUs due to decreasing the maximum sequence length and decreasing memory usage.

2.1. Model Architecture

(Figure 2) The sliding window approach mimics the architecture of a 1D convolutional neural network. The input to BERT for a single sequence is

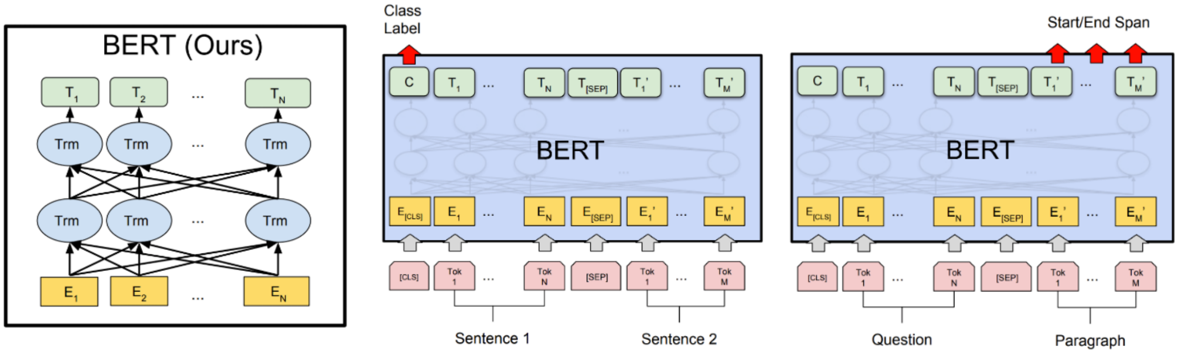
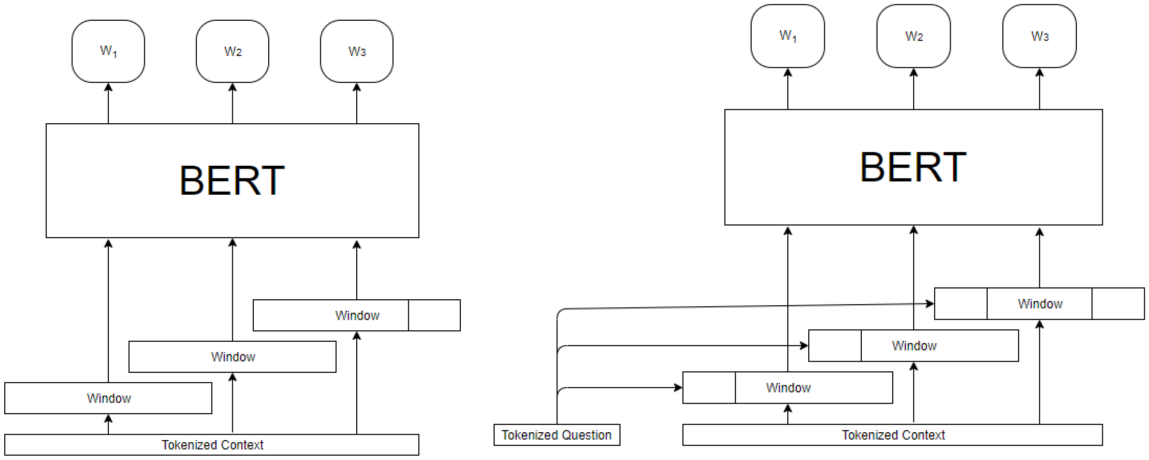


Figure 1: BERT Model



(a) Single Sequence

(b) Two Sequences

Figure 2: Sliding Window Approach Architecture

```
[CLS] SEQ Tokens [SEP]
```

where BERT transforms each of those tokens through several layers. The original classification method for a single window of BERT is to apply a linear layer to the transformed [CLS] token. For the sliding window approach, the output of each window is the transformed [CLS] token. We can take the window outputs and combine them in various ways to classify the full sequence.

2.1.1. Model Architecture with 2 Sequences

For SQuAD 2.0, we have two sequences, the question and the context. For this method, we will assume one sequence, in this case the question, is under the maximum number of tokens. The input to BERT for two sequences is

```
[CLS] SEQ1 Tokens [SEP] SEQ2 Tokens [SEP]
```

where SEQ1 is the question's tokens, and SEQ2 is the split context tokens, creating windows based on the length of the question.

2.2. Results

Merge Architecture	**SQuAD 2.0 EM**	**SQuAD 2.0 F1**	**No Ans F1**
LSTM	64.98	68.27	57.59
Mean + Linear	66.17	69.43	**59.23**
Self Attention (BERT Layer)	64.84	68.15	56.05

Table 1: Metrics of different [CLS] output vector merging approaches

Our results show that simply taking the mean of the [CLS] token outputs is sufficient in classifying long sequences of text. We use a secondary model trained on SQuAD 1.1 to answer the questions we predict to be answerable and allow us to calculate the final EM/F1 score. The score for taking the mean is above the baseline set by SQuAD 2.0 of 65/67.

3. Future Work

This research has several next steps and variables needed to be tested. The primary variable is the number of windows. A max sequence length of 384, as designated in the original BERT implementation for SQuAD, results in very few windows per example. This could be the reason why LSTMs and Self Ateention approaches to merge [CLS] output vectors do not show promising results. We should also test these methods using BERT-large because the larger model got the best results for SQuAD 1.1, and we want to be able to compare our results to Google's.

[1] P. Rajpurkar, R. Jia, P. Liang, Know what you don't know: Unanswerable questions for squad, 2018.

[2] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.